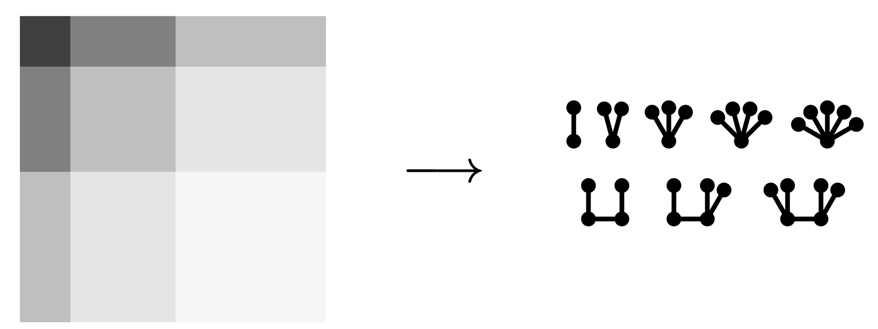
The Graph Pencil Method: Mapping Subgraph Densities to Stochastic Block Models

Lee M. Gunderson[1] Gecia Bravo-Hermsdorff[2] Peter Orbanz[1]

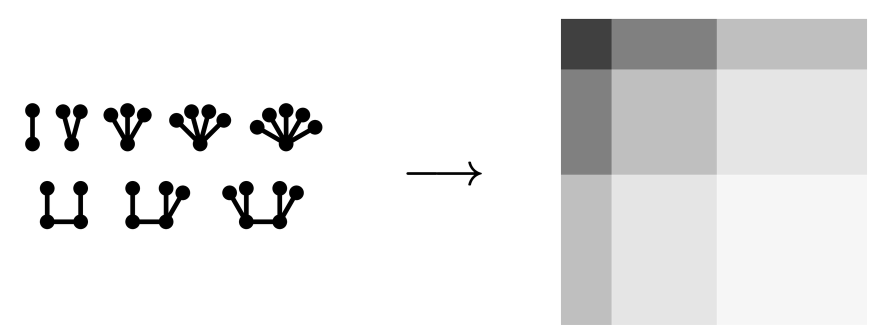
[1]Gatsby Computational Neuroscience Unit [2]Department of Statistical Science

TL;DR: Given prescribed subgraph densities, we efficiently infer a natural distribution from which sampling graphs is easy.

This direction is straightforward



This direction is rather nontrivial



Our clever method gives this inverse map!

Solves the inference and sampling problems of Exponential Random Graph Models...

- densities of small subgraphs (μ_g)
- describe the local structure ("texture")

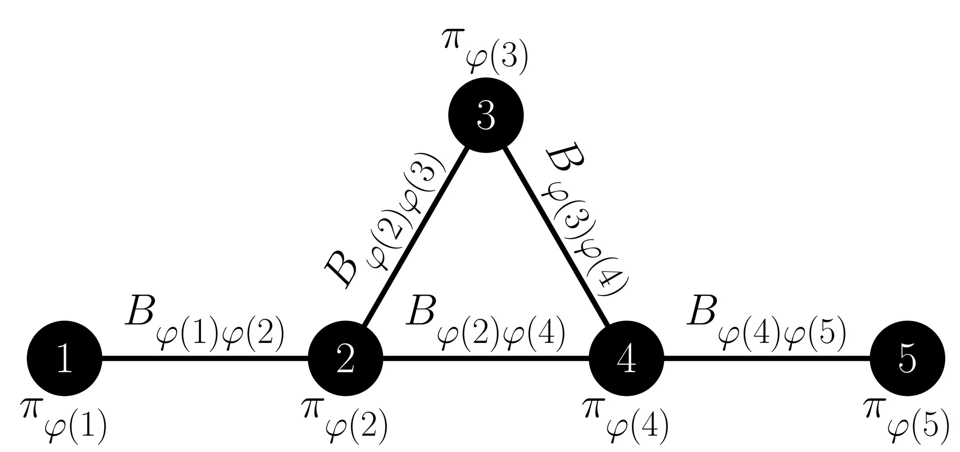
$\underbrace{(\mu_g)_{g\in\mathcal{F}\subseteq\mathcal{G}}}$ subgraph densities of a $\mathcal{F}$amily of $\mathcal{G}$raphs

...by converting them into a Stochastic Block Model!

- latent blocks/communities $(\pi_k, B_{kk'})$
- describe the global structure ("shape")

$\underbrace{(\pi_k, B_{kk'})_{k,k'\in[K]}}$ Stochastic Block Model with K communities

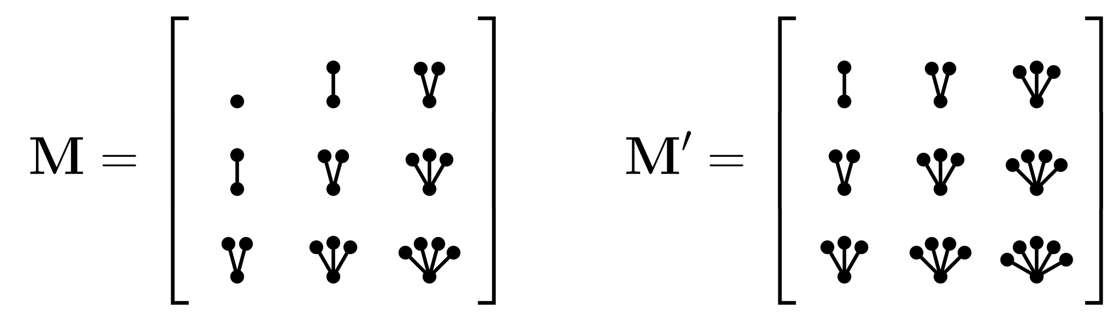
1. Subgraphs from SBMs



$$\underbrace{\mu(g)\Big|_{\mathrm{SBM}(\pi_k,B_{kk'})}}_{\text{homomorphism density } \mu \text{ of subgraph } g \text{ in an SBM given by } \pi \text{ and } \mathbf{B} \text{ with } K \text{ blocks}} = \underbrace{\sum_{\varphi:V_g\to[K]}^{K^{|V_g|}}}_{\text{sum over all maps } \varphi \text{ from vertices in } g \text{ to the } K \text{ blocks}} \left[\underbrace{\left(\prod_{i\in V_g}\pi_{\varphi(i)}\right)}_{\text{probability of assigning the vertices to those blocks}} \times \underbrace{\left(\prod_{(i,j)\in E_g} B_{\varphi(i)\varphi(j)}\right)}_{\text{probability of the corresponding edges}}\right]$$

2. Distilling the Degree Distribution

Put the star subgraph densities into two matrices:



$$\text{eigvals}\left(\mathbf{M}'\mathbf{M}^{-1}\right) = \left\{d_k\right\}$$

...then the spectrum exactly recovers the degrees of the blocks!

4. General Recipe

1. Choose a vector $\mathbf{v}$ of K or more rooted subgraphs
2. Take the outer gluing product $\mathbf{v} \circ \mathbf{v}^\top$
3. Make $\mathbf{M}$ by taking the inner product with $\boldsymbol{\pi}$
4. Make $\mathbf{M}'$ by gluing the desired rooted subgraph before taking the inner product with $\boldsymbol{\pi}$
5. The eigenvalues of $\mathbf{M}'\mathbf{M}^{-1}$ are the K latent values of the subgraph used to make $\mathbf{M}'$ in step 4

5. Proving Prony

$$\mathbf{M} = \sum_k \pi_k \begin{bmatrix} d_k^0 & d_k^1 & \cdots & d_k^{K-1} \\ d_k^1 & d_k^2 & \cdots & d_k^K \\ \vdots & \vdots & \ddots & \vdots \\ d_k^{K-1} & d_k^K & \cdots & d_k^{2K-2} \end{bmatrix} = \sum_k \pi_k \begin{bmatrix} d_k^0 \\ d_k^1 \\ \vdots \\ d_k^{K-1} \end{bmatrix} \begin{bmatrix} d_k^0 & d_k^1 & \cdots & d_k^{K-1} \end{bmatrix}$$

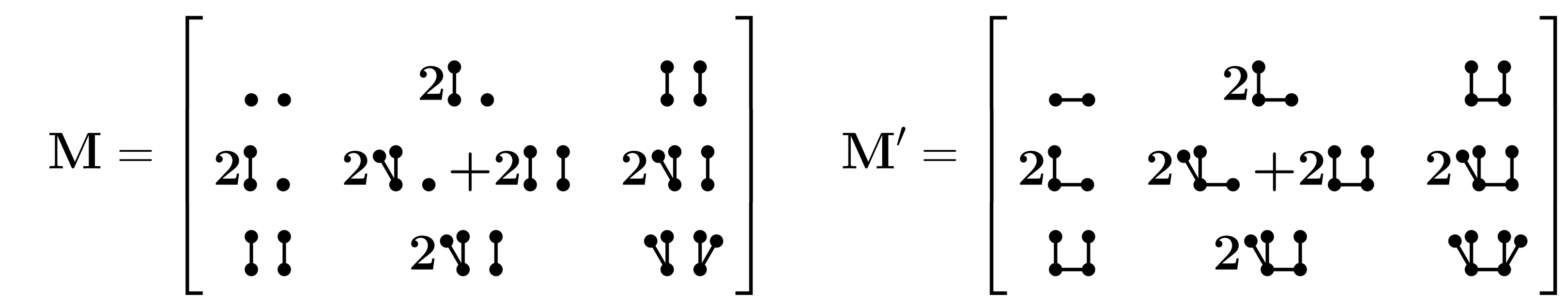
$$= \begin{bmatrix} d_1^0 & \cdots & d_K^0 \\ \vdots & \ddots & \vdots \\ d_1^{K-1} & \cdots & d_K^{K-1} \end{bmatrix} \begin{bmatrix} \pi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \pi_K \end{bmatrix} \begin{bmatrix} d_1^0 & \cdots & d_1^{K-1} \\ \vdots & \ddots & \vdots \\ d_K^0 & \cdots & d_K^{K-1} \end{bmatrix} = \mathbf{V}\,\text{diag}(\boldsymbol{\pi})\,\mathbf{V}^\top$$

$$\mathbf{M}' = \mathbf{V}\,\text{diag}(\boldsymbol{\pi}\mathbf{d})\,\mathbf{V}^\top$$

$$\mathbf{M}'\mathbf{M}^{-1} = \left(\mathbf{V}\,\text{diag}(\boldsymbol{\pi}\mathbf{d})\,\mathbf{V}^\top\right)\left(\mathbf{V}\,\text{diag}(\boldsymbol{\pi})\,\mathbf{V}^\top\right)^{-1} = \mathbf{V}\,\text{diag}(\mathbf{d})\,\mathbf{V}^{-1}$$

3. Extracting the Edge Expectations

For the edge probabilities, use the bi-star subgraph densities:

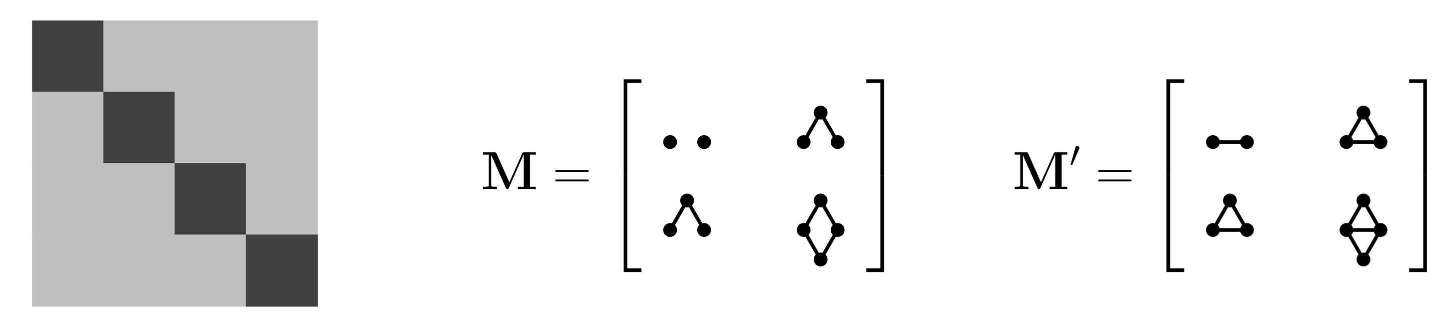


$$\text{eigvals}\left(\mathbf{M}'\mathbf{M}^{-1}\right) = \left\{B_{kk'}\right\}_{1\le k\le k'\le K}$$

...using the known eigenvectors, we can extract the entries one-by-one!

6. Considering Cycles

Even when the degrees are the same...

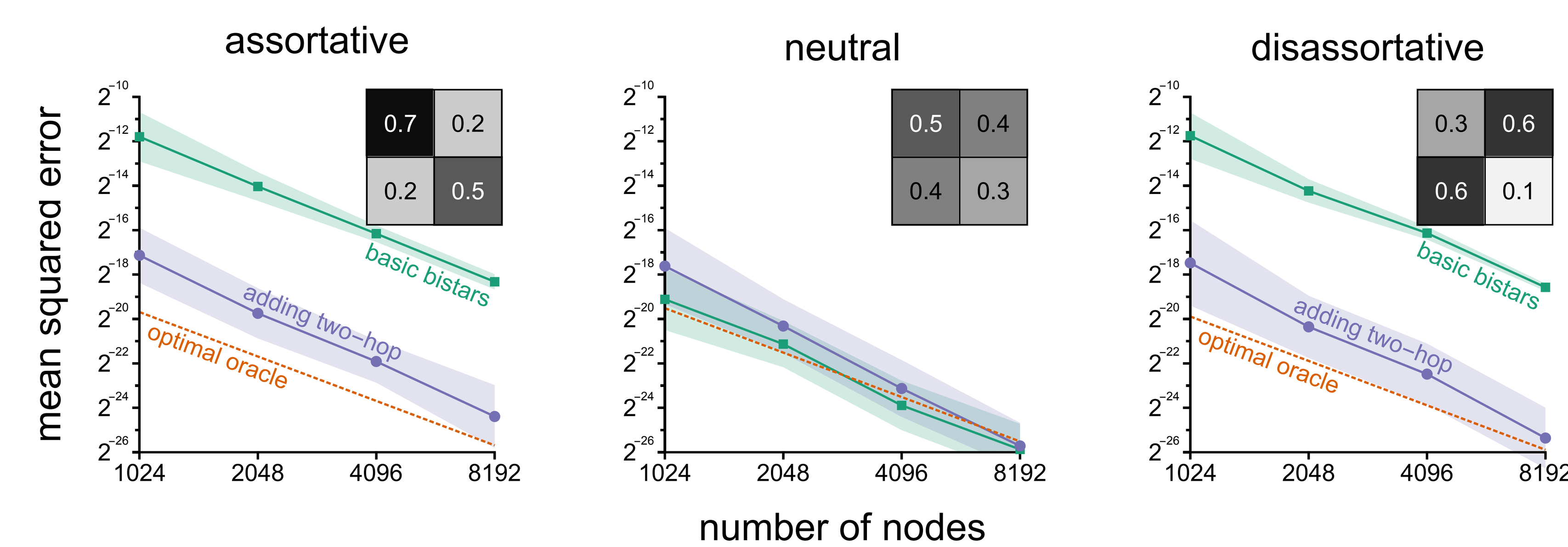


$$\text{eigvals}\left(\mathbf{M}'\mathbf{M}^{-1}\right) = \left\{B_{k=k'}, B_{k\neq k'}\right\}$$

...we can still recover the edge probabilities!

7. Some Synthetic Simulations

By adding additional subgraphs...



...we can better resolve similar degrees!